Elevate your Relationship with Data

Manpreet Singh
Lead Solution Engineer

Agenda

- Logical and Physical Data Layer with Relationship- Demo
- Flexibility and Dynamic- Demo
- Avoid Calculations through Relationship – Demo
- Tuning - Demo

Tableau Data Model, 2020.2

Tableau Until 2020.1

- Joins, De-normalized Flat Structure, **Physical**.

Tableau 2020.2

- Introduced Relationship, Logical Structure
- Logical Layer on top of Physical Layer.
- Tables remain distinct- Normalized, Native level of Detail

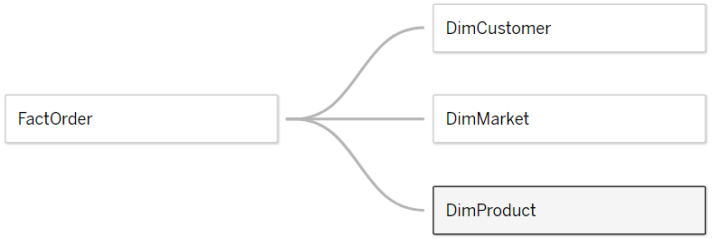

Tableau Data Model- Flexibility and Dynamic

Flexibility: Variety of Questions, Intuitive

Dynamic: JOIN is dynamically selected

Helps Analysts and IT

- Analysts don’t have to learn Data skills, which JOIN to choose to get answer for Business Query.
- Use the existing investments in IT

Example:-

Book: Granularity, One record per book id (Unique)

BookID	Title	Author ID
AD22	Alanna Saves the Day	BH149

Checkouts: Granularity, Number of time a book is checked out in a month.

Hence there are 12 records (12 months) for each Book ID.

Book ID	Checkout Month	Checkouts
AD222	1	40
AD222	3	39

Question

Which Book Titles are Checked out and How many Times ?

Which Book Titles have never been checked out ?

Physical JOIN:-

```
"SELECT
        ""Book"".""Title"" AS ""Title"",
        SUM(""Checkouts"".""Checkouts"") AS ""sum:Checkouts:ok""
FROM    ""TableauTemp"".""Checkouts$"" ""Checkouts""
 LEFT JOIN  ""TableauTemp"".""Book$"" ""Book""
        ON (""Checkouts"".""BookID"" = ""Book"".""BookID"")
GROUP BY 1“
```

++

Logical JOIN (Relationship)

```
"SELECT
        ""t0"".""Title"" AS ""Title"",
        SUM(""Checkouts"".""Checkouts"") AS ""sum:Checkouts:ok""
FROM    ""TableauTemp"".""Checkouts$"" ""Checkouts""
 INNER JOIN (
                SELECT ""Checkouts"".""BookID"" AS ""BookID (Checkouts)"",
                        ""Book"".""Title"" AS ""Title""
                FROM ""TableauTemp"".""Checkouts$"" ""Checkouts""
          LEFT JOIN  ""TableauTemp"".""Book$"" ""Book""
                ON (""Checkouts"".""BookID"" = ""Book"".""BookID"")
        GROUP BY 1,  2
        ) ""t0""
ON (""Checkouts"".""BookID"" IS NOT DISTINCT FROM ""t0"".""BookID (Checkouts)"")
GROUP BY 1"
```

Example:- Single Table View

Physical Join

```
"SELECT
        ""Book"".""Title"" AS ""Title""
FROM
        ""TableauTemp"".""Book$"" ""Book""
  FULL JOIN
        ""TableauTemp"".""Checkouts$""
        ""Checkouts""

ON (""Book"".""BookID"" = ""Checkouts"".""BookID"")

GROUP BY 1

ORDER BY ""Title"" ASC NULLS FIRST"
```

Logical JOIN (Relationship)

```
"SELECT
        ""Book"".""Title"" AS ""Title""
FROM
        ""TableauTemp"".""Book$"" ""Book""
GROUP BY 1
ORDER BY ""Title"" ASC NULLS FIRST"
```

Calculations(LoDs)

Book

BookID	Title	AuthID
TM925	The Mallemaroking	BT132

Checkouts

BookID	CheckoutMonth	Checkouts
TM925	1	121
TM925	2	113
TM925	3	112
TM925	4	126
TM925	5	126
TM925	6	121
TM925	7	130
TM925	8	107
TM925	9	81
TM925	10	77
TM925	11	69
TM925	12	60

Edition

ISBN	BookID	Format	PubID	Publication Date	Pages	Print Run Size (k)	Price
989-28-79-43574-5	TM925	Graphic	ESP	9/15/2178	187	8	23.99
989-28-79-71565-6	TM925	Hardcover	ESP	2/25/2183	819	25	21.5
989-28-79-13801-1	TM925	mass market paperback	ESP	7/19/2185	672	71	8.99
989-28-79-91028-0	TM925	trade paperback	ESP	4/14/2186	784	50	12.5

Physical JOIN

- Duplicate Records
- Total Records 48
- LoD to get correct value
- {FIXED [Title],[Checkout Month]: AVG([Checkouts (Checkouts)])}

Relationship

- No need to write LoD
- Tables are not physically joined.

Calculation LoD

Physical JOIN

```
"SELECT
        ""Book"".""BookID"" AS ""BookID"",
        SUM(""Checkouts"".""Checkouts"") AS
""sum:Checkouts:ok"",
        SUM(""Edition"".""Price"") AS ""sum:Price:ok""
FROM
        ""TableauTemp"".""Book$"" ""Book""
 INNER JOIN
        ""TableauTemp"".""Checkouts$"" ""Checkouts""
        ON (""Book"".""BookID"" = ""Checkouts"".""BookID"")
 INNER JOIN
        ""TableauTemp"".""Edition$"" ""Edition""
        ON (""Book"".""BookID"" = ""Edition"".""BookID"")
GROUP BY 1"
```

Logical JOIN (Relationship)

Query – 1

```
"SELECT
        ""t0"".""BookID"" AS ""BookID"",
         SUM(""Checkouts"".""Checkouts"") AS  "sum:Checkouts:ok""
FROM
        ""TableauTemp"".""Checkouts$"" ""Checkouts""
 LEFT JOIN (
        SELECT
                ""Book"".""BookID"" AS ""BookID""
          FROM
                ""TableauTemp"".""Book$"" ""Book""
          GROUP BY 1
        ) ""t0""
        ON (""Checkouts"".""BookID"" = ""t0"".""BookID"")
GROUP BY 1"
```

Query-2

```
"SELECT
        ""t0"".""BookID"" AS ""BookID"",
        SUM(""Edition"".""Price"") AS ""sum:Price:ok""
FROM
        ""TableauTemp"".""Edition$"" ""Edition""
 LEFT JOIN (
          SELECT
                ""Book"".""BookID"" AS ""BookID""
          FROM
                ""TableauTemp"".""Book$"" ""Book""
          GROUP BY 1
        ) ""t0""
ON (""Edition"".""BookID"" = ""t0"".""BookID"")
GROUP BY 1"
```

Optimization:- without changing cardinality and Relationship

Physical Layer (Join)

```
"SELECT
    ""DimCustomer"".""Customer Segment"" AS ""Customer Segment"",
    "DimProduct"".""Product Category"" AS ""Product Category"",
    ""DimProduct"".""Product Sub-Category"" AS ""Product Sub- Category"",
   SUM(""FactOrder"".""Profit"") AS ""sum:Profit:ok"",
   SUM(""FactOrder"".""Sales"") AS ""sum:Sales:ok""
FROM
                ""TableauTemp"".""FactOrder$"" ""FactOrder""
 INNER JOIN
                 ""TableauTemp"".""DimProduct$"" ""DimProduct""
ON (""FactOrder"".""Product Key"" = ""DimProduct"".""Product Key"")
 INNER JOIN
                ""TableauTemp"".""DimOrder$"" ""DimOrder""
ON (""FactOrder"".""Order Key"" = ""DimOrder"".""Order Key"")
 INNER JOIN
                 ""TableauTemp"".""DimMarket$"" ""DimMarket""
ON (""FactOrder"".""Market Key"" = ""DimMarket"".""Market Key"")

 INNER JOIN
                ""TableauTemp"".""DimCustomer$"" ""DimCustomer""
ON (""FactOrder"".""Customer Key"" = ""DimCustomer"".""Customer Key"")
 INNER JOIN
                ""TableauTemp"".""DimDate$"" ""DimDate""
ON (""FactOrder"".""Order Date Key"" = ""DimDate"".""Date Key"")
GROUP BY 1,
  2,
  3"
```

Logical Layer (Relationship)

```
"SELECT

    ""t0"".""Customer Segment"" AS ""Customer Segment"",

   ""t0"".""Product Category"" AS ""Product Category"",

   ""t0"".""Product Sub-Category"" AS ""Product Sub-Category"",

    SUM(""FactOrder"".""Sales"") AS ""sum:Sales:ok""

FROM

      ""TableauTeamp"".""FactOrder$"" ""FactOrder""

 INNER JOIN (

 SELECT ""FactOrder"".""Customer Key"" AS ""Customer Key"",

   ""FactOrder"".""Product Key"" AS ""Product Key"",

   ""DimCustomer"".""Customer Segment"" AS ""Customer Segment"",

   ""DimProduct"".""Product Category"" AS ""Product Category"",

   ""DimProduct"".""Product Sub-Category"" AS ""Product Sub-Category""

  FROM ""TableauTemp"".""FactOrder$"" ""FactOrder""

   LEFT JOIN ""TableauTemp"".""DimCustomer$"" ""DimCustomer"" ON
(""FactOrder"".""Customer Key"" = ""DimCustomer"".""Customer Key"")

   LEFT JOIN ""TableauTemp"".""DimProduct$"" ""DimProduct"" ON
(""FactOrder"".""Product Key"" = ""DimProduct"".""Product Key"")

  GROUP BY 1,   3,   4,   2,   5

) ""t0"" ON ((""FactOrder"".""Customer Key"" IS NOT DISTINCT FROM ""t0"".""Customer
Key"") AND (""FactOrder"".""Product Key"" IS NOT DISTINCT FROM ""t0"".""Product Key"")))

GROUP BY 1,  2,  3"
```

Optimization:- After changing Cardinality and Relationship

Logical Layer (Relationship)

```
"SELECT
    ""DimCustomer"".""Customer Segment"" AS ""Customer Segment"",
    ""DimProduct"".""Product Category"" AS ""Product Category"",
    ""DimProduct"".""Product Sub-Category"" AS ""Product Sub-Category"",
    SUM(""FactOrder"".""Profit"") AS ""sum:Profit:ok"",
    SUM(""FactOrder"".""Sales"") AS ""sum:Sales:ok""
FROM
    ""TableauTemp"".""FactOrder$"" ""FactOrder""
  INNER JOIN ""TableauTemp"".""DimCustomer$"" ""DimCustomer"" ON (""FactOrder"".""Customer Key"" = ""DimCustomer"".""Customer Key"")
  INNER JOIN ""TableauTemp"".""DimProduct$"" ""DimProduct"" ON (""FactOrder"".""Product Key"" = ""DimProduct"".""Product Key"")
GROUP BY 1,  2,  3"
```

Thank You